Mackenzie Tjogas
CMSI 402

HW Assignment #2

1.) Stephens 5.1: What's the difference between a component-based architecture and a service-oriented architecture?

In a service-oriented architecture the pieces are implemented, as the name suggests, as services. These services usually run on separate computers that talk to one another across a network, the pieces are more separate here then in a component-based architecture where pieces of the system are loosely coupled components which do services for one another.

2.) Stephens 5.2: Suppose you're building a phone application that lets you play tic-tac-toe against a simple computer opponent. It will display high scores stored on the phone, not in an external database. Which architectures would be most appropriate and why?

A simple monolithic rule-based architecture is all that would be needed for the tic-tac-toe application. Since the app is self-contained and doesn't need a databases or any remote services, a more complex architecture wouldn't be needed.

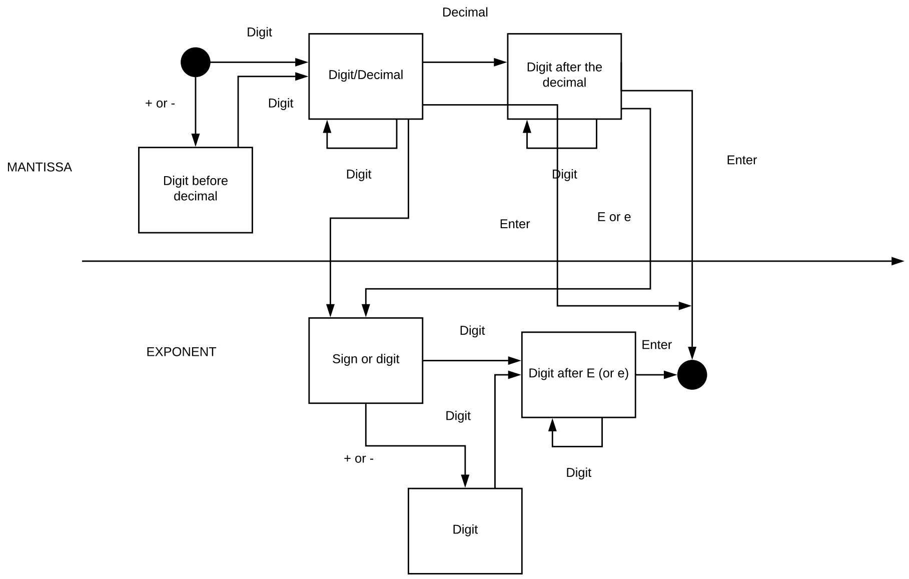
3.) Stephens 5.4: Repeat question 3 [after thinking about it; it repeats question 2 for a chess game] assuming the chess program lets two users play against each other over an Internet connection.

We could use web services here to make it so the two programs can communicate with one another over the internet, and doing so would make this chess game also run as a monolithic rule-based service-oriented application.

4.) Stephens 5.6: What kind of database structure and maintenance should the ClassyDraw application use?

ClassyDraw doesn't need a database if it stores each picture in a different file and the current drawing the user is making could be stored on a temporary file. If any emergency or data loss occurred while drawing the temporary file could be restored.

5.) Stephens 5.8: Draw a state machine diagram to let a program read floating point numbers in scientific notation as in +37 or -12.3e+17 (which means -12.3 x 1017). Allow both E and e for the exponent symbol.
[Jeez, is this like Dr. Dorin'
DFAs, or what???]



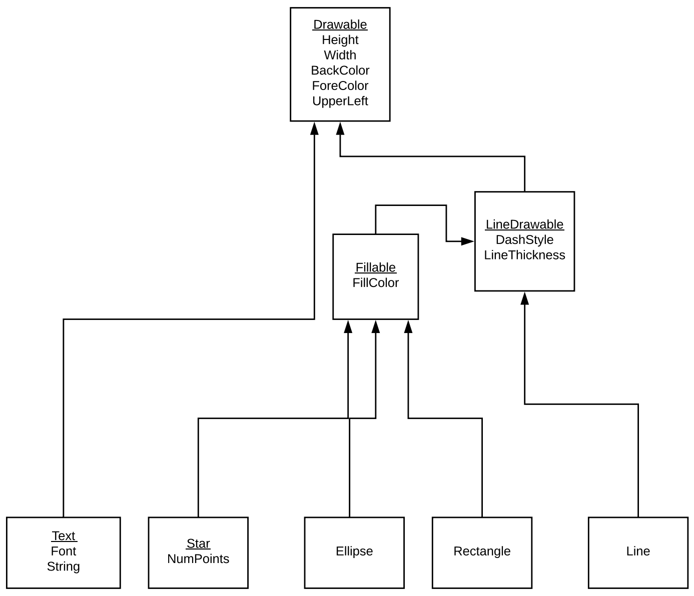
6.) Stephens 6.1: Consider the ClassyDraw classes Line, Rectangle, Ellipse, Star, and Text. What properties do these classes all share? What properties do they not share? Are there any properties shared by some classes and not others? Where should the shared and nonshared properties be implemented?

The classes share the properties needed to draw-foreground color and background color. They also all share the property that defines their position-upper-left corner, width, and height. Some of the classes though will need additional data for their shape and that data won't be shared between classes. Likewise, some properties can be shared by some classes and not others. For example: Rectangle, Ellipse, and star can all be filled so they all share the "fill color" properties.

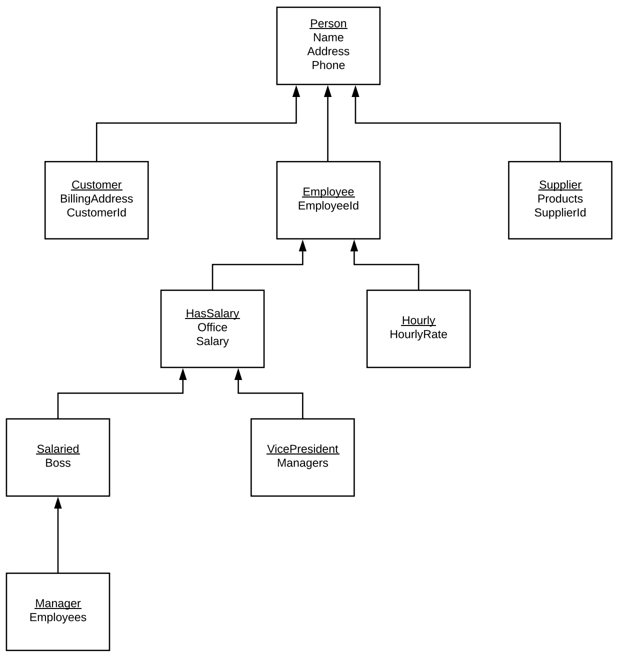
<u>Properties and who shares them:</u>
DashStyle-Rectangle, Ellipse, Line, Star
LineThickness-Rectangle, Ellipse, Line, Star
FillColor-Rectangle, Ellipse, Star
NumPoints-Star
String-Text
Font-Text
Height-All
Width-All
UpperLeft-All

BackColor-All
ForeColor-All

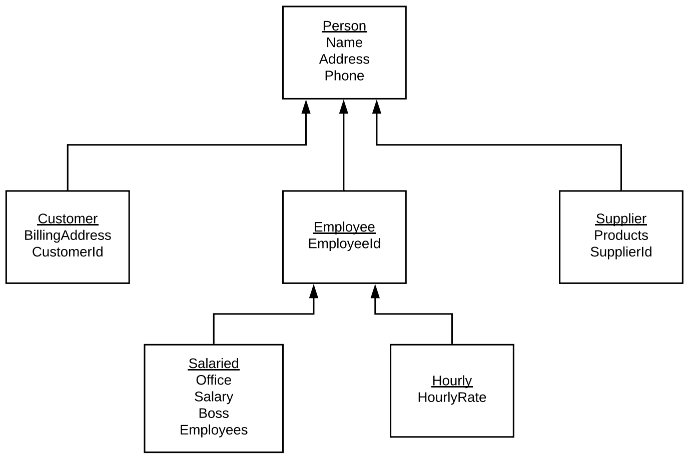
7.) Stephens 6.2: Draw an inheritance diagram showing the properties you identified for Exercise 1. (Create parent classes as needed, and don't forget the Drawable class at the top.)



8.) Stephens 6.3



9.) Stephens 6.6: Suppose your company has many managerial types such as department namager, project manager, and division manager. You also have multiple levels of vice president, some of whom reprt to other manager types. How could you combine the Salaried, Manager, and VicePresident types you used in Exercise 3? Draw the new inheritance hierarchy.


Person
Name
Address
Phone
Customer
BillingAddress
CustomerId
Employee
EmployeeId
Supplier
Products
SupplierId
Salaried
Office
Salary
Boss
Employees
Hourly
HourlyRate